

PRINCIPLE OF CONSERVATION OF STRUCTURAL IDENTITY (PCIS)

Trattato fondazionale chiuso

**Criterio necessario e sufficiente per la conservazione
dell'identità strutturale**

nei sistemi artificiali, cognitivi e auto-modificanti

Autore

Pierernesto Di Paolo

Anno

2026

**Contatto per autorizzazioni, diritti e richieste
ufficiali**

pierernesto.di.paolo.69@proton.me

PREFAZIONE

Questo lavoro nasce da una constatazione semplice e, al tempo stesso, radicale: nel dibattito contemporaneo sui sistemi cognitivi artificiali, sull'intelligenza artificiale e sull'auto-modificazione, **manca un criterio formale, necessario e sufficiente, per stabilire quando un sistema resta se stesso.**

Metriche di performance, risultati empirici, adattamento comportamentale e apprendimento statistico sono stati spesso scambiati per garanzie di identità. Non lo sono. Due sistemi possono produrre esiti indistinguibili e tuttavia essere strutturalmente diversi; viceversa, un sistema può cambiare stato, parametri e comportamento senza perdere la propria identità.

Il **Principle of Conservation of Structural Identity (PCIS)** nasce per colmare questa lacuna in modo definitivo.

Non come proposta applicativa, non come promessa tecnologica, non come filosofia dell'IA, ma come **costruzione matematica rigorosa**, fondata esclusivamente su strutture, isomorfismi e invarianti verificabili.

In questo trattato viene mostrato che:

- l'identità di un sistema può essere definita in modo formale e non ambiguo;
- tale identità è indipendente da stati, parametri ed esiti empirici;
- esiste un operatore costruttivo (OSI) che ne consente l'estrazione e la verifica;
- l'auto-modificazione è ammissibile solo se isomorfa;
- la rigenerazione è possibile, l'evoluzione strutturale no;
- la scalabilità multi-agente è ottenibile senza emergenza incontrollata;
- il controllo istituzionale è realizzabile senza discrezionalità.

Il risultato non è un sistema “più intelligente”, ma qualcosa di più raro e necessario:

un limite dimostrato, verificabile e opponibile.

Questo corpus è deliberatamente chiuso.

Ogni ulteriore sviluppo possibile è applicativo, derivato, ingegneristico, non fondazionale.

Non vengono fatte promesse di onniscienza, perfezione o autonomia illimitata: tali concetti sono esclusi non per prudenza, ma per dimostrazione.

Il lavoro è concepito per essere letto, verificato, contestato e replicato da chiunque disponga degli strumenti matematici adeguati. Non richiede fiducia, non richiede interpretazione, non richiede autorità esterna.

Richiede solo rigore.

DEDICA

Questo lavoro è dedicato a due presenze fondamentali della mia vita.

A Stellina (Catterina Giordano),

mia compagna di vita,

per la costanza, la forza silenziosa e la presenza che accompagna ogni ricerca lunga e difficile.

Senza il tuo sostegno quotidiano, nessuna costruzione, per quanto rigorosa, avrebbe trovato il tempo e lo spazio per essere portata a compimento.

A Lanfranchini Pierluigi,

mio zio materno,

che nella vita è stato per me una figura paterna, un riferimento umano e morale.

La sua influenza non è stata teorica, ma strutturale: il senso del limite, della responsabilità e della coerenza che permea questo lavoro nasce anche da ciò che mi ha trasmesso, ben prima che esistessero formule o teoremi.

Questo trattato porta con sé la traccia di entrambe queste presenze: una come fondamento affettivo, l'altra come fondamento umano.

NOTA LEGALE E COPYRIGHT

Titolarità e diritti

L'intero contenuto del presente trattato, incluse ma non limitate a:

- formulazioni teoriche,
- definizioni matematiche,
- assiomi, teoremi e dimostrazioni,
- strutture concettuali,
- applicazioni (Monitor, Rigenerazione, Multi-Agente, Audit Pack),
- architetture logiche,
- testi, schemi algoritmici, esempi implementativi,
- prefazione, dedica, note esplicative e conclusioni,

è **opera originale** ed è di **proprietà esclusiva** di:

Pierernesto Di Paolo

© 2026 — Tutti i diritti riservati

Regime giuridico

Quest'opera è rilasciata **senza alcuna licenza open**, copyleft, permissiva o implicita.

È **espressamente vietato**, senza autorizzazione scritta dell'autore:

- copiare, riprodurre o distribuire l'opera, in tutto o in parte;
- modificarla, adattarla o tradurla;
- utilizzarla come base per opere derivate;
- integrarla in prodotti software, hardware o servizi;

- impiegarla in contesti commerciali, istituzionali o di ricerca applicata;
- presentarla come propria o come lavoro collettivo;
- utilizzare concetti, strutture o risultati come “liberamente utilizzabili”.

Ogni uso non autorizzato costituisce **violazione dei diritti d'autore**.

Autorizzazioni

Ogni richiesta di utilizzo, citazione estesa, derivazione, implementazione o adozione deve essere **preventivamente autorizzata per iscritto** dall'autore.

Contatto unico per autorizzazioni:

pierernesto.di.paolo.69@proton.me

In assenza di risposta scritta positiva, **l'uso si intende negato**.

DICHIARAZIONE DI STATUTO EPISTEMOLOGICO DELL'OPERA

La presente dichiarazione definisce in modo vincolante la natura, lo statuto epistemologico e i limiti del corpus PCIS.

Questa opera è:

- **fondazionale**, non applicativa
- **teorica**, ma con dimostrazioni operative
- **chiusa**, non estendibile
- **non ideologica**
- **non predittiva**

Non è:

- un manuale di implementazione,
- un progetto industriale,
- una proposta normativa,
- una promessa di intelligenza generale.

Il suo scopo è unico e preciso:

**stabilire un criterio necessario e sufficiente
per la conservazione dell'identità strutturale
sotto trasformazioni interne ed esterne.**

Il PCIS opera a un livello **strutturale**, non fenomenologico.

Non valuta:

- prestazioni,

- esiti,
- efficienza,
- correttezza funzionale,
- bontà etica.

Valuta esclusivamente:

- **la struttura formale che definisce l'identità del sistema.**

Questo rende il PCIS:

- indipendente dalla tecnologia,
- indipendente dal dominio,
- indipendente dall'implementazione fisica.

Un sistema biologico, artificiale, simbolico o ibrido è valutabile **solo** in quanto istanza di una struttura.

Il PCIS non compete con:

- teorie dell'apprendimento,
- modelli cognitivi,
- architetture neurali,
- sistemi multi-agente esistenti.

Opera **a monte**.

Così come:

- la teoria della computabilità non dice *come* programmare,
- ma cosa è computabile,

il PCIS non dice *come* costruire un sistema,
ma **quando quel sistema smette di essere se stesso**.

Il corpus PCIS è **formalmente chiuso**.

Questo significa che:

- non verranno introdotti nuovi assiomi fondazionali,
- non verranno estesi i principi qui definiti,
- non verrà modificato il criterio centrale.

Ogni lavoro futuro potrà solo:

- applicare,
- interpretare,
- criticare,
- confrontare,

ma **non estendere** il nucleo teorico.

Questa chiusura non costituisce una limitazione del quadro teorico, ma una garanzia di stabilità, verificabilità e non ambiguità nel tempo.

Il testo è organizzato in modo da essere:

- leggibile sequenzialmente,
- verificabile per parti,
- utilizzabile come riferimento canonico.

Le sezioni centrali introducono:

- il modello strutturale,
- l'operatore di identità,
- il criterio di equivalenza,
- le trasformazioni ammissibili.
-

Le sezioni successive dimostrano il principio tramite **quattro progetti operativi replicabili**, che costituiscono prova concreta e indipendente della validità del PCIS.

Le appendici contengono:

- pseudo-codice,
- formati dati,
- casi di test,
- criteri di audit,
- istruzioni di replicabilità.

1. FONDAMENTI FORMALI E MOTIVAZIONE STRUTTURALE

Questa sezione introduce le fondamenta matematiche minime necessarie per l'intero trattato.

Ogni nozione utilizzata nelle sezioni successive è **definita qui**.

Non verranno introdotti concetti non formalizzati in questa sezione o in quelle immediatamente successive.

1.1 Principio metodologico di minimalità strutturale

Il framework PCIS adotta il seguente principio metodologico vincolante.

Principio 1.1 (Minimalità strutturale)

Una struttura è ammessa nel formalismo se e solo se:

- è definibile come insieme finito o combinazione finita di insiemi,
- è dotata esclusivamente di relazioni esplicite,
- non richiede semantica esterna per essere interpretata,
- è verificabile mediante procedura deterministica.

Questo principio esclude:

- strutture infinite non costruttive,
- continuità topologiche non necessarie,
- probabilità come primitive,
- nozioni di “grado”, “intensità” o “peso” non formalizzabili.

1.2 Universo formale e tipi

Si assume un contesto matematico **costruttivo e tipato**.

Definizione 1.1 (Tipo)

Un tipo è un insieme non vuoto dotato di una specificazione formale che ne delimita l'uso operativo.

I tipi sono trattati come insiemi etichettati, non come classi informali.

Indicazione:

- i tipi sono indicati con lettere maiuscole latine (T, A, C, ...),
- non è ammesso l'uso di tipi impliciti.

1.3 Strutture matematiche finite

Definizione 1.2 (Struttura)

Una struttura è una tupla finita:

$$S = \langle U, R_1, \dots, R_k, F_1, \dots, F_m \rangle$$

dove:

- U è un insieme di base finito (universo),
- R_i sono relazioni su U,
- F_j sono funzioni su U.

Ogni struttura è completamente determinata dai suoi componenti.

Non è ammessa alcuna interpretazione esterna.

1.4 Relazioni e ordini

Definizione 1.3 (Relazione)

Una relazione R su un insieme U è un sottoinsieme di $U \times U$.

Definizione 1.4 (Ordine parziale)

Una relazione $\leq$ su U è un ordine parziale se è:

- riflessiva,
- antisimmetrica,
- transitiva.

Definizione 1.5 (Ordine totale)

Un ordine parziale $\leq$ è totale se:

per ogni $x, y \in U$ vale:

$x \leq y$ oppure $y \leq x$.

1.5 Isomorfismo strutturale

La nozione di identità in PCIS **non è uguaglianza sintattica**, ma isomorfismo.

Definizione 1.6 (Isomorfismo strutturale)

Siano:

$$S1 = \langle U1, R1^1, \dots, F1^1, \dots \rangle$$

$$S2 = \langle U2, R1^2, \dots, F1^2, \dots \rangle$$

Un isomorfismo tra $S1$ e $S2$ è una funzione biiettiva:

$$\varphi : U1 \rightarrow U2$$

tale che:

- per ogni relazione R :
 $(x, y) \in R1^1 \Leftrightarrow (\varphi(x), \varphi(y)) \in R1^2$
- per ogni funzione F :
 $\varphi(F1^1(x1, \dots, xn)) = F1^2(\varphi(x1), \dots, \varphi(xn))$

Conseguenza fondamentale

Due strutture isomorfe sono **strutturalmente identiche**, anche se:

- hanno etichette diverse,
- hanno implementazioni diverse,
- producono esiti diversi.

1.6 Principio di non-compensabilità

Il PCIS esclude **per costruzione** ogni riduzione numerica compensativa.

Definizione 1.7 (Vincolo non compensabile)

Sia $(U, \leq)$ una struttura ordinata.

L'ordine è non compensabile se **non esiste** alcuna funzione:

$$f : U \rightarrow \mathbb{R}$$

tale che:

$$x \leq y \Leftrightarrow f(x) \leq f(y)$$

In altre parole, l'ordine non è rappresentabile da una funzione scalare.

Questo vincolo:

- esclude pesi,
- esclude utilità,
- esclude aggregazioni lineari o non lineari.

1.7 Motivazione della scelta strutturale

Il formalismo PCIS **non utilizza**:

- metriche continue,
- probabilità,
- funzioni di costo,

- funzioni di utilità.

Motivo:

ogni tentativo di rappresentare l'identità tramite numeri introduce **compensabilità**, e quindi **ambiguità strutturale**.

1.8 Ruolo dei fondamenti nel trattato

Le definizioni di questa sezione sono sufficienti per:

- definire ruoli funzionali come strutture tipate,
- definire obiettivi come poset non compensabili,
- definire gerarchie decisionali totali,
- definire funzioni di valore senza numericità,
- definire identità come classe di isomorfismo.

Nessuna sezione successiva utilizzerà concetti non introdotti qui.

1.9 Nota di formalizzabilità (proof-grade)

Tutte le definizioni di questa sezione sono:

- traducibili in logica del primo ordine,
- formalizzabili in un proof assistant (es. Lean),
- decidibili per strutture finite.

Non è richiesto alcun assioma non costruttivo.

2. RUOLI FUNZIONALI (COMPONENTE F)

Questa sezione definisce in modo **matematicamente completo** cosa sia un ruolo funzionale, quali siano i suoi invarianti strutturali e quando due insiemi di ruoli debbano essere considerati identici dal punto di vista dell'identità strutturale del sistema.

2.1 Motivazione formale

Ogni sistema artificiale, naturale o ibrido può essere descritto come insieme di **ruoli** che:

- ricevono input,
- producono output,
- occupano una posizione funzionale nel sistema.

Il PCIS **non identifica i ruoli con agenti, moduli, persone o processi**, ma con **strutture tipate**.

Questo evita ogni ambiguità semantica.

2.2 Spazio dei tipi

Definizione 2.1 (Spazio dei tipi)

Sia T un insieme non vuoto di tipi, come definiti nella Sezione 1.2.

I tipi rappresentano domini e codomini ammissibili per funzioni operative all'interno di un sistema.

Esempi (non vincolanti):

- dati,
- segnali,
- stati simbolici,

- configurazioni strutturate.

Il formalismo **non assume alcuna semantica** dei tipi.

2.3 Definizione formale di ruolo funzionale

Definizione 2.2 (Ruolo funzionale)

Un ruolo funzionale è una tripla tipata:

$$r = \langle I_r, O_r, \sigma_r \rangle$$

dove:

- $I_r \in T$ è il tipo di input del ruolo,
- $O_r \in T$ è il tipo di output del ruolo,
- σ_r è un identificatore strutturale appartenente a un insieme finito ΣF .

L'identificatore σ_r :

- non è semantico,
- non è un nome,
- non è una funzione,
- è una pura etichetta strutturale formale.

2.4 Insieme dei ruoli funzionali

Definizione 2.3 (Insieme dei ruoli)

L'insieme dei ruoli funzionali di un sistema è un insieme finito:

$$F = \{ r_1, r_2, \dots, r_m \}$$

dove ogni r_i è un ruolo funzionale ai sensi della Definizione 2.2.

Non sono ammessi:

- ruoli duplicati,

- ruoli privi di tipizzazione,
- ruoli con identificatore strutturale ambiguo.

2.5 Isomorfismo tra insiemi di ruoli

L'identità strutturale **non dipende dai nomi** ma dalla struttura.

Definizione 2.4 (Isomorfismo di ruoli)

Siano:

$$F1 = \{ r1^1, \dots, rn^1 \}$$

$$F2 = \{ r1^2, \dots, rn^2 \}$$

due insiemi di ruoli funzionali.

Un isomorfismo di ruoli è una biiezione:

$$\varphi : F1 \rightarrow F2$$

tale che, per ogni $r \in F1$ con:

$$r = \langle Ir, Or, \sigma r \rangle$$

si abbia:

$$\varphi(r) = \langle Ir, Or, \sigma r \rangle$$

In altre parole, l'isomorfismo preserva:

- il tipo di input,
- il tipo di output,
- l'identificatore strutturale.

2.6 Invarianti strutturali dei ruoli

Definizione 2.5 (Invarianti strutturali di F)

Gli invarianti strutturali dell'insieme dei ruoli F sono:

- la cardinalità $|F|$,

- il multinsieme delle coppie (Ir, Or),
- l'insieme degli identificatori strutturali $\{ \sigma r \mid r \in F \}$.

Qualsiasi trasformazione che alteri almeno uno di questi elementi:
modifica la struttura di F.

2.7 Identità strutturale dei ruoli

Definizione 2.6 (Classe di isomorfismo di F)

L'identità strutturale dell'insieme dei ruoli è la classe:

$$[F] = \{ F' \mid F' \cong F \}$$

Due sistemi che condividono la stessa classe [F]:

- sono indistinguibili dal punto di vista dei ruoli funzionali,
- anche se implementati in modo differente.

2.8 Separazione netta da concetti informali

Nel PCIS:

- un ruolo non è un agente,
- un ruolo non è una persona,
- un ruolo non è un modulo software,
- un ruolo non è una funzione computazionale concreta.

Un ruolo è **una posizione strutturale tipata.**

2.9 Osservazioni costruttive

- Tutte le strutture sono finite.
- L'isomorfismo è decidibile.

- La definizione è direttamente formalizzabile in Lean.
- Nessuna semantica esterna è richiesta.

2.10 Nota di chiusura della sezione

La componente F cattura **cosa può fare il sistema**, ma **non perché** né **con quale priorità**.

Questi aspetti saranno formalizzati separatamente nelle componenti G e H.

3. OBIETTIVI FONDAMENTALI

(Componente G dello spazio strutturale)

Questa sezione formalizza la componente **G**, che rappresenta **gli obiettivi fondamentali** del sistema.

Gli obiettivi non sono utilità, non sono punteggi, non sono funzioni di costo: sono **vincoli strutturali**.

3.1 Ruolo strutturale degli obiettivi

Nel framework PCIS, gli obiettivi:

- non sono numerici,
- non sono aggregabili,
- non sono compensabili,
- non sono riducibili a funzioni scalari.

Essi definiscono **cosa non può essere sacrificato** durante l'evoluzione del sistema.

3.2 Insieme degli obiettivi fondamentali

Definizione 3.1 (Insieme degli obiettivi)

Sia O un insieme finito non vuoto.

Gli elementi di O rappresentano **obiettivi fondamentali** del sistema.

Gli elementi di O :

- sono atomi strutturali,
- non sono numeri,
- non sono pesi,

- non sono preferenze.

3.3 Relazione di priorità parziale

Gli obiettivi possono essere in relazione di precedenza **strutturale**, non quantitativa.

Definizione 3.2 (Relazione di priorità parziale)

Una relazione $\leq \subseteq O \times O$ è detta relazione di priorità se soddisfa:

Riflessività

Per ogni $o \in O$:

$$o \leq o$$

Antisimmetria

Per ogni $o_1, o_2 \in O$:

$$(o_1 \leq o_2 \wedge o_2 \leq o_1) \Rightarrow o_1 = o_2$$

Transitività

Per ogni $o_1, o_2, o_3 \in O$:

$$(o_1 \leq o_2 \wedge o_2 \leq o_3) \Rightarrow o_1 \leq o_3$$

La coppia $(O, \leq)$ costituisce un **insieme parzialmente ordinato (poset)**.

3.4 Interpretazione strutturale dell'ordine

L'ordine $\leq$ **non rappresenta preferenze**.

In particolare:

- $o_1 \leq o_2$ significa che **o_1 non può essere sacrificato per realizzare o_2** ;
- l'assenza di relazione tra o_1 e o_2 indica **incomparabilità strutturale**, non indifferenza.

Questo esclude ogni schema multicriterio compensativo.

3.5 Reticolo degli obiettivi (caso opzionale)

Definizione 3.3 (Reticolo degli obiettivi)

Il poset $(O, \leq)$ è un reticolo se per ogni $o_1, o_2 \in O$ esistono:

- un estremo inferiore $o_1 \wedge o_2$,
- un estremo superiore $o_1 \vee o_2$.

La struttura di reticolo:

- **non è obbligatoria**,
- ma quando presente è un **invariante strutturale osservabile**.

3.6 Isomorfismo tra strutture di obiettivi

L'identità degli obiettivi non dipende dai nomi, ma dalla struttura di priorità.

Definizione 3.4 (Isomorfismo di obiettivi)

Siano:

- $(O_1, \leq_1)$,
- $(O_2, \leq_2)$

due strutture di obiettivi.

Un isomorfismo è una biiezione:

$$\varphi : O_1 \rightarrow O_2$$

tale che, per ogni $o_1, o_2 \in O_1$:

$$o_1 \leq_1 o_2$$

se e solo se

$$\varphi(o_1) \leq_2 \varphi(o_2)$$

3.7 Invarianti strutturali degli obiettivi

Definizione 3.5 (Invarianti strutturali di G)

Gli invarianti strutturali della componente G sono:

- la cardinalità $|O|$;
- la struttura del poset $(O, \preceq)$ a isomorfismo;
- l'eventuale struttura di reticolo.

Qualsiasi trasformazione che alteri uno di questi elementi:
modifica l'identità strutturale di G.

3.8 Identità strutturale degli obiettivi

Definizione 3.6 (Classe di isomorfismo di G)

L'identità strutturale della componente degli obiettivi è la classe:

$$[G] = \{ (O', \preceq') \mid (O', \preceq') \cong (O, \preceq) \}$$

Due sistemi che condividono la stessa classe $[G]$:

- hanno gli stessi vincoli fondamentali,
- anche se implementati in modo completamente diverso.

3.9 Non-compensabilità strutturale degli obiettivi

Il formalismo esclude **per costruzione** la riduzione a funzioni di utilità.

Non esiste alcuna funzione:

$$f : O \rightarrow \mathbb{R}$$

tale che:

$$o_1 \preceq o_2 \Leftrightarrow f(o_1) \leq f(o_2)$$

Questo rende impossibile ogni compensazione numerica.

3.10 Osservazioni costruttive

- Il poset è finito.
- L'isomorfismo è decidibile.
- La struttura è formalizzabile in Lean.
- Nessuna probabilità, metrica o semantica esterna è utilizzata.

3.11 Nota di chiusura della sezione

La componente G cattura **cosa il sistema non può violare**.

La decisione su *come* scegliere tra alternative compatibili sarà formalizzata nella componente H.

4. GERARCHIA DECISIONALE NON COMPENSABILE

(Componente H dello spazio strutturale)

Questa sezione formalizza la componente **H**, che rappresenta la **gerarchia decisionale fondamentale** del sistema.

Essa stabilisce **come** vengono risolti i conflitti tra alternative ammissibili **senza mai ricorrere a compensazioni numeriche**.

4.1 Ruolo strutturale della gerarchia decisionale

Nel framework PCIS, la gerarchia decisionale:

- non è una funzione di utilità,
- non è un ranking numerico,
- non è un peso,
- non è una media.

Essa è un **vincolo strutturale assoluto** che stabilisce precedenze non negoziabili.

4.2 Insieme dei criteri decisionali

Definizione 4.1 (Insieme dei criteri)

Sia C un insieme finito non vuoto.

Gli elementi di C rappresentano **criteri decisionali fondamentali** del sistema.

I criteri:

- sono atomi strutturali,
- non ammettono pesi,

- non ammettono aggregazioni,
- non ammettono compensazioni.

4.3 Ordine totale sui criteri

Definizione 4.2 (Ordine totale)

Una relazione $\leq_H \subseteq C \times C$ è un ordine totale se soddisfa:

Riflessività

Per ogni $c \in C$:

$$c \leq_H c$$

Antisimmetria

Per ogni $c_1, c_2 \in C$:

$$(c_1 \leq_H c_2 \wedge c_2 \leq_H c_1) \Rightarrow c_1 = c_2$$

Transitività

Per ogni $c_1, c_2, c_3 \in C$:

$$(c_1 \leq_H c_2 \wedge c_2 \leq_H c_3) \Rightarrow c_1 \leq_H c_3$$

Totalità

Per ogni $c_1, c_2 \in C$:

$$c_1 \leq_H c_2 \text{ oppure } c_2 \leq_H c_1$$

La coppia $(C, \leq_H)$ costituisce un **insieme totalmente ordinato**.

4.4 Non-compensabilità formale della gerarchia

La non-compensabilità è qui definita **in senso matematico**, non semantico.

Definizione 4.3 (Non-compensabilità)

L'ordine totale $(C, \leq_H)$ è non compensabile se **non esiste alcuna funzione**:

$$f: C \rightarrow R$$

tale che:

$$c_1 \leq_H c_2 \Leftrightarrow f(c_1) \leq f(c_2)$$

e tale funzione sia utilizzabile per definire:

- combinazioni lineari,
- somme pesate,
- funzioni aggregative.

In particolare, non è ammessa alcuna funzione:

$$F : \mathbb{R}^n \rightarrow \mathbb{R}$$

che consenta di sacrificare un criterio c_1 per migliorare un criterio c_2 quando:

$$c_1 <_H c_2$$

4.5 Interpretazione strutturale dell'ordine

Se:

$$c_1 <_H c_2$$

allora:

- nessuna realizzazione di c_2 può giustificare la violazione di c_1 ;
- l'ordine è un **vincolo**, non una preferenza.

Questa proprietà distingue H **radicalmente** da ogni schema multicriterio standard.

4.6 Isomorfismo tra gerarchie decisionali

Definizione 4.4 (Isomorfismo di gerarchie)

Siano:

- $(C_1, \leq_{H_1})$,
- $(C_2, \leq_{H_2})$

due gerarchie decisionali.

Un isomorfismo è una biiezione:

$$\varphi : C1 \rightarrow C2$$

tale che, per ogni $c1, c2 \in C1$:

$$c1 \leq_{H1} c2$$

se e solo se

$$\varphi(c1) \leq_{H2} \varphi(c2)$$

L'isomorfismo preserva l'ordine totale completo.

4.7 Invarianti strutturali della gerarchia

Definizione 4.5 (Invarianti strutturali di H)

Gli invarianti strutturali della componente H sono:

- la cardinalità $|C|$;
- la struttura dell'ordine totale a isomorfismo;
- la non-compensabilità dell'ordine.

Qualsiasi trasformazione che:

- introduca parzialità,
- consenta compensazione,
- modifichi l'ordine,

altera l'identità strutturale di H.

4.8 Identità strutturale della gerarchia

Definizione 4.6 (Classe di isomorfismo di H)

L'identità strutturale della gerarchia decisionale è la classe:

$$[H] = \{ (C', \leq_{H'}) \mid (C', \leq_{H'}) \cong (C, \leq_H) \}$$

Due sistemi che condividono la stessa classe [H]:

- sono indistinguibili dal punto di vista decisionale,
- anche se implementati con logiche completamente diverse.

4.9 Separazione netta dagli obiettivi

La componente H:

- **non sostituisce** gli obiettivi G,
- **non li ordina quantitativamente**,
- stabilisce solo **come decidere** quando più alternative rispettano G.

G = vincoli fondamentali

H = precedenze decisionali assolute

4.10 Osservazioni costruttive

- L'ordine totale è decidibile.
- La non-compensabilità è un vincolo logico.
- Nessuna rappresentazione numerica è ammessa.
- La struttura è formalizzabile direttamente in Lean.

4.11 Nota di chiusura della sezione

La componente H impedisce **per costruzione**:

- derive utilitaristiche,
- compromessi numerici,
- ottimizzazioni ingannevoli.

Essa è il cuore del vincolo decisionale del PCIS.

5. FUNZIONE DI VALORE

(Componente V dello spazio strutturale)

Questa sezione formalizza la componente V , chiarendo in modo rigoroso **cosa costituisce** la funzione di valore di un sistema e **cosa non ne fa parte**.

Il punto centrale è la **separazione netta tra forma funzionale e parametri**.

5.1 Ruolo strutturale della funzione di valore

Nel framework PCIS, la funzione di valore:

- **non** è una funzione di utilità classica,
- **non** è un punteggio numerico,
- **non** è un criterio di ottimizzazione globale.

Essa è una **struttura di valutazione** che associa alternative a esiti secondo una **forma funzionale invariabile**, separata da ogni adattamento parametrico.

5.2 Spazio delle alternative

Definizione 5.1 (Spazio delle alternative)

Sia A un insieme non vuoto.

Gli elementi di A rappresentano **alternative decisionali astratte**.

Non si assume:

- struttura numerica su A ,
- metrica,
- ordinamento totale.

5.3 Spazio degli esiti valutativi

Definizione 5.2 (Spazio degli esiti)

Sia D un insieme non vuoto.

Gli elementi di D rappresentano **esiti valutativi**.

Su D può essere definita:

- una relazione di confronto,
- eventualmente parziale.

Non è richiesta numericità.

5.4 Funzione di valore parametrizzata

Definizione 5.3 (Funzione di valore)

Una funzione di valore è una funzione:

$$v : A \times \Theta \rightarrow D$$

dove:

- Θ è uno spazio di parametri,
- D è lo spazio degli esiti.

I parametri Θ :

- possono variare nel tempo,
- rappresentano stato, contesto o adattamento,
- **non fanno parte dell'identità strutturale.**

5.5 Forma funzionale della funzione di valore

Definizione 5.4 (Forma funzionale)

La forma funzionale di una funzione di valore v è la struttura:

$$\text{form}(v) = \langle A, D, C, R \rangle$$

dove:

- A è lo spazio delle alternative,
- D è lo spazio degli esiti,
- C è l'insieme dei criteri utilizzati nella valutazione,
- R è l'insieme delle relazioni strutturali tra criteri ed esiti.

I parametri Θ **non appartengono** alla forma funzionale.

5.6 Assioma di separazione forma-parametri

Assioma 5.1 (Separazione forma-parametri)

Due funzioni di valore v_1 e v_2 possono differire nei parametri Θ senza differire nella forma funzionale.

Formalmente:

$$\text{form}(v_1) = \text{form}(v_2)$$

anche se

$$v_1 \neq v_2$$

Questo assioma è essenziale per distinguere:

- adattamento legittimo,
- mutazione strutturale.

5.7 Isomorfismo tra forme funzionali

Definizione 5.5 (Isomorfismo di forme funzionali)

Siano:

$$\text{form}(v_1) = \langle A_1, D_1, C_1, R_1 \rangle$$

$$\text{form}(v_2) = \langle A_2, D_2, C_2, R_2 \rangle$$

Un isomorfismo di forma è una coppia di biiezioni:

$$\varphi_A : A_1 \rightarrow A_2$$

$$\varphi_D : D_1 \rightarrow D_2$$

tale che:

- φ_A e φ_D preservino le relazioni in R ,
- i criteri in C_1 corrispondano biunivocamente a quelli in C_2 .

Se tale isomorfismo esiste, si scrive:

$$\text{form}(v_1) \cong \text{form}(v_2)$$

5.8 Invarianti strutturali della funzione di valore

Definizione 5.6 (Invarianti strutturali di V)

Gli invarianti strutturali della componente V sono:

- la cardinalità e tipologia dei criteri C ;
- la struttura delle relazioni R ;
- la tipologia dello spazio degli esiti D ;
- la separazione formale tra forma e parametri.

Qualsiasi trasformazione che alteri uno di questi elementi:
modifica l'identità strutturale di V .

5.9 Identità strutturale della funzione di valore

Definizione 5.7 (Classe di isomorfismo di V)

L'identità strutturale della funzione di valore è la classe:

$$[V] = \{ v' \mid \text{form}(v') \cong \text{form}(v) \}$$

Due sistemi che condividono la stessa classe $[V]$:

- valutano alternative nello stesso modo strutturale,

- anche se producono esiti diversi per parametri diversi.

5.10 Riflesso della non-compensabilità

La non-compensabilità della gerarchia H si riflette in V tramite:

- relazioni R non aggregative,
- assenza di combinazioni lineari sui criteri,
- assenza di scalarizzazione.

Questo garantisce coerenza globale tra H e V .

5.11 Osservazioni costruttive

- Nessuna assunzione di numericità è necessaria.
- Gli isomorfismi sono verificabili per strutture finite.
- La definizione è direttamente formalizzabile in Lean.
- La funzione di valore non introduce semantica nascosta.

5.12 Nota di chiusura della sezione

La componente V cattura **come il sistema valuta**, ma **non cosa deve essere preservato** (G) né **quali precedenze sono assolute** (H).

Questa separazione è cruciale per evitare derive concettuali.

6. STRUTTURA COMPLETA DEL SISTEMA

(Il sistema strutturale Σ)

Questa sezione integra le componenti formalizzate nelle Sezioni 2–5 in **un'unica struttura matematica composta**, che rappresenta l'identità strutturale completa di un sistema nel framework PCIS.

6.1 Motivazione dell'integrazione strutturale

Le componenti:

- F (ruoli funzionali),
- G (obiettivi fondamentali),
- H (gerarchia decisionale),
- V (funzione di valore),

sono state definite separatamente per evitare ambiguità concettuali.

Questa sezione mostra come esse:

- si combinano in modo **non ridondante**,
- preservano le rispettive identità,
- formano una struttura globale coerente.

6.2 Definizione formale di sistema strutturale

Definizione 6.1 (Sistema strutturale)

Un sistema strutturale è una quaterna:

$$\Sigma = \langle F, G, H, V \rangle$$

dove:

- F è un insieme finito di ruoli funzionali (Sezione 2),
- $G = (O, \leq)$ è una struttura di obiettivi fondamentali parzialmente ordinata (Sezione 3),
- $H = (C, \leq_H)$ è una gerarchia decisionale totalmente ordinata e non compensabile (Sezione 4),
- V è una funzione di valore con forma funzionale definita (Sezione 5).

Ogni componente è una **struttura formale autonoma**, dotata dei propri invarianti.

6.3 Spazio dei sistemi strutturali

Definizione 6.2 (Spazio dei sistemi)

Sia S l'insieme di tutti i sistemi strutturali ammissibili:

$$S = \{ \Sigma = \langle F, G, H, V \rangle \mid \\ F, G, H, V \text{ soddisfano le definizioni precedenti} \}$$

Lo spazio S :

- non è metrico continuo,
- non è probabilistico,
- non è topologico.

È uno spazio **puramente strutturale**.

6.4 Isomorfismo tra sistemi strutturali

L'identità strutturale globale non è uguaglianza sintattica, ma **isomorfismo composto**.

Definizione 6.3 (Isomorfismo globale)

Siano:

$$\Sigma_1 = \langle F_1, G_1, H_1, V_1 \rangle$$

$$\Sigma_2 = \langle F_2, G_2, H_2, V_2 \rangle$$

Un isomorfismo strutturale globale è una quaterna:

$$\Phi = (\varphi_F, \varphi_G, \varphi_H, \varphi_V)$$

tale che:

- $\varphi_F : F_1 \rightarrow F_2$ è un isomorfismo di ruoli (Def. 2.4),
- $\varphi_G : G_1 \rightarrow G_2$ è un isomorfismo di strutture di obiettivi (Def. 3.4),
- $\varphi_H : H_1 \rightarrow H_2$ è un isomorfismo di gerarchie decisionali (Def. 4.4),
- $\varphi_V : \text{form}(V_1) \rightarrow \text{form}(V_2)$ è un isomorfismo di forme funzionali (Def. 5.5).

Se tale Φ esiste, si scrive:

$$\Sigma_1 \cong \Sigma_2$$

6.5 Identità strutturale globale

Definizione 6.4 (Identità strutturale del sistema)

L'identità strutturale di un sistema Σ è la sua classe di isomorfismo globale:

$$[\Sigma] = \{ \Sigma' \in S \mid \Sigma' \cong \Sigma \}$$

Due sistemi appartengono alla stessa identità strutturale **se e solo se** sono isomorfi in tutte le componenti.

6.6 Invarianti strutturali globali

Definizione 6.5 (Invarianti globali)

Gli invarianti strutturali globali di un sistema Σ sono:

$$\text{Inv}(\Sigma) = \{ [F], [G], [H], [V] \}$$

Qualsiasi trasformazione che alteri almeno uno di questi elementi:

modifica l'identità strutturale globale del sistema.

6.7 Distinzione tra struttura e stato

È fondamentale distinguere:

- **Struttura:** la quaterna Σ e la sua classe $[\Sigma]$;
- **Stato:** configurazioni contingenti, parametriche o temporanee.

Assioma 6.1 (Separazione stato–struttura)

Variazioni di stato che non alterano alcun invariante strutturale non modificano l'identità del sistema.

Questo assioma formalizza la distinzione tra:

- adattamento legittimo,
- mutazione strutturale.

6.8 Conseguenze immediate

Dal formalismo segue che:

- un sistema può cambiare comportamento senza cambiare identità;
- un sistema può cambiare identità senza rivelarlo immediatamente nel comportamento;
- l'identità non è inferibile dagli esiti.

6.9 Osservazioni costruttive

- L'isomorfismo globale è decidibile per strutture finite.
- La definizione è composizionale.
- Ogni componente è verificabile separatamente.
- L'intera struttura è formalizzabile in Lean.

6.10 Nota di chiusura della sezione

La quaterna Σ è l'**oggetto centrale del PCIS**.

Tutto ciò che segue opererà **su Σ** , sulle sue trasformazioni e sulla sua identità.

7. FIRMA STRUTTURALE CANONICA

(Spazio Ω e rappresentazione normalizzata dell'identità)

Questa sezione introduce la **firma strutturale canonica**, che costituisce una rappresentazione **normalizzata, verificabile e completa** dell'identità strutturale di un sistema nel framework PCIS.

La firma è progettata per essere:

- costruttiva,
- invariabile per isomorfismo,
- completa (necessaria e sufficiente),
- computabile per strutture finite.

7.1 Motivazione della firma strutturale

L'identità strutturale di un sistema Σ è definita come classe di isomorfismo $[\Sigma]$. Tuttavia, una classe di isomorfismo è un oggetto **astratto**.

La firma strutturale fornisce:

- un **rappresentante canonico** di $[\Sigma]$,
- un oggetto **confrontabile per uguaglianza sintattica**,
- una base per definire operatori decisionali e verifiche automatiche.

7.2 Spazio delle firme strutturali

Definizione 7.1 (Spazio delle firme)

Sia Ω l'insieme delle firme strutturali, definite come tuple finite di invarianti normalizzati.

Formalmente:

$$\Omega = \Omega_F \times \Omega_G \times \Omega_H \times \Omega_V$$

dove:

- Ω_F è lo spazio delle firme dei ruoli funzionali,
- Ω_G è lo spazio delle firme delle strutture di obiettivi,
- Ω_H è lo spazio delle firme delle gerarchie decisionali,
- Ω_V è lo spazio delle firme delle forme funzionali di valore.

7.3 Firma dei ruoli funzionali

Definizione 7.2 (Firma della componente F)

La firma dei ruoli funzionali è la coppia:

$$\sigma_F = \langle |F|, MF \rangle$$

dove:

- $|F|$ è la cardinalità dell'insieme dei ruoli,
- MF è il multinsieme delle coppie tipate (Ir, Or) , considerato a isomorfismo.

Due insiemi di ruoli hanno la stessa firma se e solo se:

$$[F1] = [F2].$$

7.4 Firma della struttura degli obiettivi

Definizione 7.3 (Firma della componente G)

La firma della struttura degli obiettivi è la coppia:

$$\sigma G = \langle |O|, \text{Poset}(O, \leq) \rangle$$

dove:

- $|O|$ è la cardinalità dell'insieme degli obiettivi,
- $\text{Poset}(O, \leq)$ è la classe di isomorfismo del poset degli obiettivi.

Se il poset è un reticolo, tale informazione è incorporata nella classe di isomorfismo.

7.5 Firma della gerarchia decisionale

Definizione 7.4 (Firma della componente H)

La firma della gerarchia decisionale è la tripla:

$$\sigma H = \langle |C|, \text{Order}(C, \leq_H), NC \rangle$$

dove:

- $|C|$ è la cardinalità dei criteri decisionali,
- $\text{Order}(C, \leq_H)$ è la classe di isomorfismo dell'ordine totale,
- NC è un flag strutturale che attesta la non-compensabilità.

7.6 Firma della funzione di valore

Definizione 7.5 (Firma della componente V)

La firma della funzione di valore è definita come:

$$\sigma V = \text{form}(V)$$

dove $\text{form}(V)$ è la forma funzionale definita nella Sezione 5, considerata a isomorfismo.

I parametri della funzione di valore **non sono inclusi** nella firma.

7.7 Firma strutturale canonica globale

Definizione 7.6 (Firma strutturale canonica)

La firma strutturale canonica di un sistema:

$$\Sigma = \langle F, G, H, V \rangle$$

è la tupla:

$$\text{Sig}(\Sigma) = \langle \sigma_F, \sigma_G, \sigma_H, \sigma_V \rangle \in \Omega$$

7.8 Invarianza per isomorfismo

Proposizione 7.1 (Invarianza)

Se:

$$\Sigma_1 \cong \Sigma_2$$

allora:

$$\text{Sig}(\Sigma_1) = \text{Sig}(\Sigma_2)$$

Dimostrazione.

Ogni componente della firma è definita come classe di isomorfismo della rispettiva struttura. ■

7.9 Completezza della firma

Proposizione 7.2 (Completezza)

Se:

$$\text{Sig}(\Sigma_1) = \text{Sig}(\Sigma_2)$$

allora:

$$\Sigma_1 \cong \Sigma_2$$

Dimostrazione.

L'uguaglianza delle firme implica l'uguaglianza delle classi di isomorfismo di tutte le componenti.

7.10 Teorema di unicità canonica

Teorema 7.1 (Unicità a isomorfismo)

La firma strutturale $\text{Sig}(\Sigma)$ è unica a meno di isomorfismo del sistema Σ .

La firma è quindi una **rappresentazione canonica completa** dell'identità strutturale.

7.11 Osservazioni costruttive

- La firma è finita per strutture finite.
- Il confronto tra firme è decidibile.
- La firma è indipendente da implementazione e parametri.
- La costruzione è formalizzabile direttamente in Lean.

7.12 Nota di chiusura della sezione

La firma strutturale canonica è il **ponte operativo** tra:

- identità strutturale astratta,
- verifica concreta e automatizzabile.

Su di essa sarà definito l'operatore OSI.

8. OPERATORE OSI

(Operational Structural Identity – definizione costruttiva)

Questa sezione definisce l'operatore **OSI** come funzione **costruttiva, non circolare e computabile** che associa a ogni sistema strutturale la sua **firma strutturale canonica**.

OSI non introduce nuovi concetti:

è un **operatore di estrazione normalizzata** dell'identità strutturale.

8.1 Motivazione dell'operatore OSI

L'identità strutturale $[\Sigma]$ è una classe di isomorfismo.

La firma $\text{Sig}(\Sigma)$ è una rappresentazione canonica.

L'operatore OSI:

- rende l'identità **operativamente accessibile**,
- consente confronto **per uguaglianza sintattica**,
- elimina ambiguità semantiche,
- impedisce definizioni circolari.

8.2 Dominio e codominio

Definizione 8.1 (Dominio)

Il dominio dell'operatore OSI è lo spazio dei sistemi strutturali:

$$\text{Dom}(\text{OSI}) = S$$

come definito nella Sezione 6.2.

Definizione 8.2 (Codominio)

Il codominio dell'operatore OSI è lo spazio delle firme strutturali:

$$\text{Cod}(\text{OSI}) = \Omega$$

come definito nella Sezione 7.1.

8.3 Definizione costruttiva di OSI

Definizione 8.3 (Operatore OSI)

L'operatore OSI è la funzione:

$$\text{OSI} : S \rightarrow \Omega$$

definita, per ogni sistema:

$$\Sigma = \langle F, G, H, V \rangle$$

come:

$$\text{OSI}(\Sigma) = \text{Sig}(\Sigma)$$

dove $\text{Sig}(\Sigma)$ è la firma strutturale canonica costruita nella Sezione 7.

Esplicitamente:

$$\text{OSI}(\Sigma) = \langle \sigma_F(F), \sigma_G(G), \sigma_H(H), \sigma_V(V) \rangle$$

Questa definizione è **non circolare**:

- OSI è definito *a partire* dalle strutture,
- non definisce le strutture.

8.4 OSI come funzione computabile

Proposizione 8.1 (Computabilità)

Se il sistema Σ è finito (cioè F, O, C, A sono insiemi finiti), allora $\text{OSI}(\Sigma)$ è computabile in tempo finito.

Dimostrazione.

Ogni componente della firma:

- conta cardinalità finite,
- costruisce classi di isomorfismo su strutture finite,
- verifica proprietà decidibili (ordini, relazioni).

Pertanto OSI è computabile per costruzione.

8.5 Correttezza dell'operatore OSI

Teorema 8.1 (Correttezza)

Per ogni sistema strutturale Σ :

$$\text{OSI}(\Sigma) \in \Omega$$

ed è una firma strutturale valida.

Dimostrazione.

Segue direttamente dalla costruzione di ciascuna componente della firma.

8.6 Completezza dell'operatore OSI

Teorema 8.2 (Completezza)

Per ogni $\Sigma_1, \Sigma_2 \in S$:

$$\text{OSI}(\Sigma_1) = \text{OSI}(\Sigma_2)$$

se e solo se

$$\Sigma_1 \cong \Sigma_2$$

Dimostrazione.

Per le Proposizioni 7.1 e 7.2, la firma canonica è invariabile e completa rispetto all'isomorfismo globale.

8.7 Assenza di semantica nascosta

OSI **non**:

- valuta performance,
- valuta outcome,
- valuta utilità,
- utilizza metriche numeriche.

OSI opera **esclusivamente** su invarianti strutturali.

Ogni informazione non strutturale è **per costruzione ignorata**.

8.8 Invarianza rispetto a stato e parametri

Proposizione 8.2 (Invarianza parametrica)

Sia Σ' ottenuto da Σ variando esclusivamente:

- parametri della funzione di valore V ,
- stati contingenti del sistema.

Allora:

$$\text{OSI}(\Sigma') = \text{OSI}(\Sigma)$$

Dimostrazione.

I parametri non fanno parte della forma funzionale né di alcuna firma strutturale.

8.9 Interpretazione operativa

OSI può essere interpretato come:

- estrattore di identità strutturale,
- normalizzatore canonico a isomorfismo,
- funzione di “hash strutturale” non crittografico.

Questa interpretazione è **operativa**, non semantica.

8.10 Osservazioni costruttive

- OSI è totale su S.
- OSI è deterministico.
- OSI è verificabile.
- OSI è formalizzabile direttamente in Lean.

8.11 Nota di chiusura della sezione

OSI è il **cardine operativo del PCIS**.

Ogni nozione di equivalenza, distanza e trasformazione farà riferimento esclusivo a OSI.

9. EQUIVALENZA STRUTTURALE E DISTANZA DISCRETA

Questa sezione introduce una nozione **verificabile** di equivalenza strutturale tra sistemi e una **distanza discreta** coerente con la natura qualitativa dell'identità strutturale nel framework PCIS.

9.1 Motivazione

Nel PCIS l'identità:

- **non ammette gradazioni,**
- **non è continua,**
- **non è probabilistica.**

Due sistemi:

- o condividono la stessa identità strutturale,
- o non la condividono.

Ne segue che ogni nozione di “distanza” deve essere **binaria**.

9.2 Equivalenza strutturale

Definizione 9.1 (Equivalenza strutturale)

Siano $\Sigma_1, \Sigma_2 \in S$.

Si dice che Σ_1 e Σ_2 sono **strutturalmente equivalenti** se:

$$\Sigma_1 \sim \Sigma_2$$

se e solo se

$$OSI(\Sigma_1) = OSI(\Sigma_2)$$

Per il Teorema 8.2, questa condizione equivale a:

$$\Sigma_1 \cong \Sigma_2$$

9.3 Proprietà dell'equivalenza

Proposizione 9.1

La relazione $\sim$ è una relazione di equivalenza su S .

Dimostrazione.

Riflessività:

$$\text{OSI}(\Sigma) = \text{OSI}(\Sigma)$$

Simmetria:

$$\text{se } \text{OSI}(\Sigma_1) = \text{OSI}(\Sigma_2), \text{ allora } \text{OSI}(\Sigma_2) = \text{OSI}(\Sigma_1)$$

Transitività:

$$\text{se } \text{OSI}(\Sigma_1) = \text{OSI}(\Sigma_2) \text{ e } \text{OSI}(\Sigma_2) = \text{OSI}(\Sigma_3), \text{ allora } \text{OSI}(\Sigma_1) = \text{OSI}(\Sigma_3)$$

9.4 Classi di equivalenza

Definizione 9.2 (Classe di equivalenza strutturale)

La classe di equivalenza strutturale di un sistema Σ è l'insieme:

$$[\Sigma]_{\sim} = \{ \Sigma' \in S \mid \Sigma' \sim \Sigma \}$$

Questa classe coincide con la classe di isomorfismo globale definita in Sezione 6.4.

9.5 Distanza strutturale discreta

Poiché l'identità strutturale è binaria, si introduce una distanza **discreta**, non continua.

Definizione 9.3 (Distanza strutturale)

La distanza strutturale è la funzione:

$$d : S \times S \rightarrow \{0, 1\}$$

definita come:

$$d(\Sigma 1, \Sigma 2) = 0 \text{ se } \text{OSI}(\Sigma 1) = \text{OSI}(\Sigma 2)$$

$$d(\Sigma 1, \Sigma 2) = 1 \text{ altrimenti}$$

9.6 Proprietà metriche minime

Proposizione 9.2

La funzione d soddisfa le proprietà minime di una metrica discreta:

Non negatività:

$$d(\Sigma 1, \Sigma 2) \geq 0$$

Identità degli indiscernibili:

$$d(\Sigma 1, \Sigma 2) = 0 \Leftrightarrow \Sigma 1 \sim \Sigma 2$$

Simmetria:

$$d(\Sigma 1, \Sigma 2) = d(\Sigma 2, \Sigma 1)$$

Disuguaglianza triangolare:

$$d(\Sigma 1, \Sigma 3) \leq d(\Sigma 1, \Sigma 2) + d(\Sigma 2, \Sigma 3)$$

Dimostrazione.

Segue direttamente dalla definizione su $\{0,1\}$.

9.7 Interpretazione strutturale della distanza

La distanza d **non misura quanto** due sistemi differiscono.

Essa misura **se** differiscono strutturalmente.

- $d = 0$: identità strutturale conservata
- $d = 1$: identità strutturale violata

Non esistono valori intermedi.

9.8 Robustezza rispetto a stato e parametri

Proposizione 9.3

Variazioni di stato o parametri che non alterano la firma strutturale non modificano la distanza:

$$d(\Sigma, \Sigma') = 0$$

Dimostrazione.

Segue dalla Proposizione 8.2.

9.9 Conseguenze operative

Dal formalismo segue che:

- non esiste drift strutturale “graduale”;
- ogni mutazione identitaria è rilevabile;
- non è possibile “avvicinarsi” a un’altra identità senza attraversare una discontinuità.

9.10 Osservazioni costruttive

- L’equivalenza è decidibile.
- La distanza è computabile.

- La distanza è indipendente da implementazione.
- Il formalismo è direttamente formalizzabile in Lean.

9.11 Nota di chiusura della sezione

La distanza strutturale completa il quadro concettuale:

- identità come classe,
- equivalenza come uguaglianza di firma,
- distanza come violazione binaria.

Questo prepara la definizione rigorosa delle **trasformazioni strutturali**.

10. TRASFORMAZIONI STRUTTURALI E AMMISSIBILITÀ

Questa sezione formalizza le **trasformazioni strutturali** come funzioni sullo spazio dei sistemi e distingue in modo rigoroso tra:

- trasformazioni **ammissibili** (che preservano l'identità),
- trasformazioni **non ammissibili** (che la violano).

Il criterio è **unico, necessario e sufficiente**: l'operatore OSI.

10.1 Trasformazioni come funzioni sui sistemi

Definizione 10.1 (Trasformazione strutturale)

Una trasformazione strutturale è una funzione:

$$T : S \rightarrow S$$

che associa a ogni sistema strutturale Σ un sistema trasformato $T(\Sigma)$.

Non si assume:

- continuità,
- reversibilità,
- probabilità,
- dinamica temporale.

T è un oggetto puramente strutturale.

10.2 Decomposizione analitica delle trasformazioni

Ogni trasformazione T può essere analizzata (non necessariamente in modo unico) come:

$$T(\Sigma) = \langle TF(F), TG(G), TH(H), TV(V) \rangle$$

dove:

- TF agisce sui ruoli funzionali,
- TG sugli obiettivi,
- TH sulla gerarchia decisionale,
- TV sulla funzione di valore.

Questa decomposizione è **analitica**, non operativa.

10.3 Trasformazioni ammissibili

Definizione 10.2 (Ammissibilità)

Una trasformazione $T : S \rightarrow S$ è **ammissibile** se e solo se:

$$\text{OSI}(T(\Sigma)) = \text{OSI}(\Sigma)$$

per ogni Σ nel dominio di interesse.

Equivalentemente:

$$d(\Sigma, T(\Sigma)) = 0$$

10.4 Caratterizzazione delle trasformazioni ammissibili

Proposizione 10.1

Una trasformazione T è ammissibile se e solo se ciascuna trasformazione locale:

TF, TG, TH, TV

preserva la **classe di isomorfismo** della rispettiva componente.

Dimostrazione.

Segue dalla definizione della firma strutturale come tupla di firme componenti (Sezione 7).

10.5 Trasformazioni non ammissibili

Definizione 10.3 (Non ammissibilità)

Una trasformazione T è **non ammissibile** se esiste almeno un sistema Σ tale che:

$$\text{OSI}(T(\Sigma)) \neq \text{OSI}(\Sigma)$$

Equivalentemente:

$$d(\Sigma, T(\Sigma)) = 1$$

10.6 Esempi strutturali astratti

Sono **non ammissibili** trasformazioni che:

- alterano la cardinalità di F ;
- modificano la struttura di ordine di G ;
- introducono compensabilità in H ;
- cambiano la forma funzionale di V .

Sono **ammissibili** trasformazioni che:

- variano parametri;
- cambiano stati contingenti;
- rietichettano elementi a isomorfismo.

10.7 Composizione di trasformazioni

Definizione 10.4 (Composizione)

Date due trasformazioni:

$$T_1, T_2 : S \rightarrow S$$

la loro composizione è:

$$(T_2 \circ T_1)(\Sigma) = T_2(T_1(\Sigma))$$

10.8 Chiusura sotto composizione

Proposizione 10.2 (Chiusura)

Se T_1 e T_2 sono trasformazioni ammissibili, allora $T_2 \circ T_1$ è ammissibile.

Dimostrazione.

$$\text{OSI}(T_1(\Sigma)) = \text{OSI}(\Sigma)$$

$$\text{OSI}(T_2(T_1(\Sigma))) = \text{OSI}(T_1(\Sigma))$$

Per transitività:

$$\text{OSI}(T_2(T_1(\Sigma))) = \text{OSI}(\Sigma)$$

10.9 Trasformazione identità

Definizione 10.5 (Identità)

La trasformazione identità:

$$\text{id} : S \rightarrow S$$

$$\text{id}(\Sigma) = \Sigma$$

è ammissibile.

10.10 Struttura algebrica delle trasformazioni

L'insieme delle trasformazioni ammissibili:

- contiene l'identità,
- è chiuso sotto composizione.

Pertanto forma un **monoide**.

10.11 Conseguenze fondamentali

Dal formalismo segue che:

- ogni evoluzione legittima è tracciabile;

- non esiste mutazione identitaria “silenziosa”;
- l’ammissibilità è decidibile.

10.12 Osservazioni costruttive

- L’ammissibilità è verificabile tramite OSI.
- Le trasformazioni sono finite e computabili.
- La struttura è formalizzabile in Lean.

10.13 Nota di chiusura della sezione

Le trasformazioni strutturali costituiscono il **meccanismo dinamico** del PCIS.
La sezione successiva dimostrerà risultati **non tautologici** su queste trasformazioni.

11. TEOREMI FONDAMENTALI

(Risultati non tautologici del framework PCIS)

Questa sezione presenta i **teoremi centrali** del PCIS nella loro forma **formalmente rigorosa**.

I risultati non ripetono definizioni: introducono **contenuto informativo reale**, escludendo derive strutturali non osservabili e criteri decisionali inadeguati.

11.1 Proprietà strutturali dimostrabili

Per evitare ogni circolarità, si chiarisce cosa significhi “proprietà dimostrabile”.

Definizione 11.1 (Proprietà strutturale dimostrabile)

Una proprietà P è detta **strutturale** se dipende esclusivamente dalla firma strutturale:

$P(\Sigma)$ dipende solo da $OSI(\Sigma)$.

Una proprietà strutturale è **dimostrabile** se esiste una dimostrazione formale che utilizza esclusivamente:

- le definizioni delle Sezioni 1–10,
- le regole logiche standard.

11.2 Preservazione delle proprietà dimostrabili

Teorema 11.1 (Preservazione)

Sia $T : S \rightarrow S$ una trasformazione ammissibile.

Per ogni sistema Σ e per ogni proprietà strutturale dimostrabile P :

$P(\Sigma) \Rightarrow P(T(\Sigma))$.

Dimostrazione.

Poiché T è ammissibile, $OSI(T(\Sigma)) = OSI(\Sigma)$.

Dato che P dipende solo da OSI , segue che P vale anche per $T(\Sigma)$.

Contenuto informativo.

Le dimostrazioni strutturali sono **invarianti** sotto ogni evoluzione ammessa, indipendentemente da stati e parametri.

11.3 Impossibilità del drift strutturale non osservabile

Questo risultato esclude formalmente la possibilità di mutazioni identitarie “silenziose”.

Definizione 11.2 (Drift strutturale)

Un drift strutturale è una sequenza di trasformazioni:

$$\Sigma_0 \rightarrow \Sigma_1 \rightarrow \dots \rightarrow \Sigma_n$$

tale che:

$$\Sigma_0 \neq \Sigma_n.$$

Il drift è **non osservabile** se per ogni k :

$$\text{OSI}(\Sigma_k) = \text{OSI}(\Sigma_{k-1}).$$

Teorema 11.2 (Impossibilità del drift non osservabile)

Non esiste alcuna sequenza di trasformazioni che realizzi un drift strutturale non osservabile.

Dimostrazione.

Se per ogni k vale $\text{OSI}(\Sigma_k) = \text{OSI}(\Sigma_{k-1})$, allora per chiusura delle trasformazioni ammissibili:

$$\text{OSI}(\Sigma_n) = \text{OSI}(\Sigma_0).$$

Per il Teorema 8.2 segue $\Sigma_n \cong \Sigma_0$, in contraddizione con la definizione di drift.

Contenuto informativo.

Il teorema esclude **in modo assoluto** ogni slittamento identitario graduale non rilevato.

11.4 Necessità dell'ancoraggio strutturale

Questo risultato mostra che OSI non è una scelta arbitraria, ma **necessaria**.

Definizione 11.3 (Criterio basato su esito)

Un criterio decisionale è detto basato su esito se dipende esclusivamente da:

- performance,
- outcome,
- metriche numeriche,
- valutazioni empiriche,

e non dalla firma strutturale.

Teorema 11.3 (Insufficienza dei criteri basati su esito)

Nessun criterio decisionale basato esclusivamente su esiti può garantire la conservazione dell'identità strutturale.

Dimostrazione (schema).

Esistono sistemi Σ_1 e Σ_2 tali che:

- producono esiti indistinguibili rispetto a qualunque criterio basato su esito,
- ma $OSI(\Sigma_1) \neq OSI(\Sigma_2)$.

Pertanto tali criteri non discriminano identità strutturali diverse.

Contenuto informativo.

Il teorema dimostra la **necessità logica** di un criterio strutturale come PCIS.

11.5 Chiusura del nucleo teorico

Corollario 11.1 (Completezza del nucleo PCIS)

Il framework PCIS fornisce:

- un criterio necessario e sufficiente per l'identità strutturale,

- una procedura verificabile (OSI),
- una classe completa di trasformazioni ammissibili.

Non sono richiesti ulteriori assiomi per stabilire l'identità strutturale.

11.6 Osservazioni costruttive

- Tutti i teoremi sono formalizzabili in Lean.
- Nessun risultato utilizza assunzioni empiriche.
- Ogni teorema ha contenuto informativo non nullo.
- Il nucleo teorico è **chiuso** rispetto allo scopo dichiarato.

11.7 Nota di chiusura della sezione

I teoremi del PCIS:

- non ottimizzano,
- non predicono,
- non valutano esiti,

ma **stabiliscono confini matematici inviolabili** per l'identità strutturale.

12. DISCUSSIONE FORMALE ED EPISTEMOLOGICA

(Natura, portata e limiti del framework PCIS)

Questa sezione chiarisce **cosa il PCIS garantisce, cosa non pretende di garantire e perché tale delimitazione è necessaria** dal punto di vista matematico ed epistemologico.

12.1 Indipendenza da outcome e performance

Il formalismo PCIS è **strutturalmente indipendente** da:

- esiti empirici,
- metriche di performance,
- valutazioni quantitative,
- risultati osservabili ex post.

Questa indipendenza **non è una debolezza**, ma una conseguenza necessaria dei risultati dimostrati nella Sezione 11.

In particolare:

- due sistemi possono essere indistinguibili rispetto a qualunque criterio basato su esiti,
- pur avendo identità strutturali differenti.

Ne segue che **nessun criterio basato su outcome** può fungere da garanzia di conservazione identitaria.

12.2 Dominio di validità del formalismo

Il PCIS è **dominio-indipendente**.

Il formalismo:

- non assume una specifica architettura computazionale,
- non assume una tecnologia,
- non assume un algoritmo,
- non assume una semantica applicativa.

Le uniche entità utilizzate sono:

- insiemi finiti,
- relazioni,
- ordini,
- isomorfismi,
- funzioni costruttive.

Questa generalità è **matematica**, non una pretesa applicativa.

12.3 Distinzione formale tra identità e comportamento

Il framework stabilisce una distinzione netta tra:

- **identità strutturale**: la classe di isomorfismo $[\Sigma]$;
- **comportamento osservabile**: funzione di stati, parametri e contingenze.

Conseguenze fondamentali:

- il comportamento può variare senza mutare l'identità;
- l'identità può mutare senza manifestarsi immediatamente nel comportamento.

Questa separazione evita:

- false garanzie di sicurezza,
- confusione tra adattamento e mutazione,
- valutazioni retroattive basate sugli esiti.

12.4 Limiti dichiarati del PCIS

Il PCIS **non pretende** di:

- prevedere comportamenti futuri,
- garantire correttezza funzionale,
- garantire sicurezza operativa,
- garantire optimalità,
- sostituire valutazioni empiriche o normative.

Il suo scopo è **unico e preciso**:

fornire un criterio matematicamente rigoroso, necessario e sufficiente, per stabilire se l'identità strutturale di un sistema è stata conservata.

Ogni estensione oltre questo ambito è epistemologicamente indebita.

12.5 Relazione con criteri empirici e normativi

Il PCIS **non sostituisce**:

- metriche di performance,
- test empirici,
- valutazioni normative, etiche o giuridiche.

Il PCIS **li precede logicamente**.

Ogni valutazione basata su esiti è affidabile **solo dopo** che:

- l'identità strutturale è stata verificata,
- oppure vincolata tramite OSI.

In assenza di tale verifica, nessuna valutazione ex post è logicamente fondata.

12.6 Robustezza e inevitabilità del framework

Dalle sezioni precedenti emerge che il PCIS è:

- robusto rispetto a variazioni parametriche,
- immune al drift strutturale non osservabile,
- chiuso sotto trasformazioni ammissibili,
- completo rispetto allo scopo dichiarato.

Queste proprietà:

- non dipendono da scelte tecnologiche,
- non dipendono da assunzioni empiriche,
- non dipendono da interpretazioni semantiche.

Dipendono **solo** dalla struttura matematica introdotta.

12.7 Posizionamento epistemologico finale

Il PCIS:

- non è una teoria del comportamento,
- non è una teoria dell'intelligenza,
- non è una teoria dell'ottimizzazione.

È una **teoria dei limiti strutturali dell'identità**.

In questo senso, il suo ruolo è fondazionale e **non competitivo** rispetto ad altri approcci.

12.8 Nota di chiusura della sezione

Questa discussione delimita in modo esplicito:

- il potere del formalismo,
- i suoi confini,
- la sua inevitabilità logica.

La sezione finale dichiarerà formalmente lo **stato di completezza** dell'opera.

13. CONCLUSIONE MATEMATICA E STATO FINALE DELL'OPERA

(Chiusura canonica del corpus PCIS)

Questa sezione conclude formalmente il trattato **PCIS – Principle of Conservation of Structural Identity**, dichiarando in modo esplicito **cosa è stato dimostrato, con quali strumenti, e in quali limiti**, nonché lo **stato finale e chiuso dell'opera**.

13.1 Riassunto dei risultati formali ottenuti

Nel corso delle Sezioni 1–12 è stata costruita una formalizzazione matematica completa e costruttiva del framework PCIS. In particolare:

1. È stata definita una classe di **sistemi strutturali** come quaterne:

$$\Sigma = \langle F, G, H, V \rangle$$

con componenti formalizzate tramite:

- insiemi finiti,
 - relazioni,
 - ordini,
 - isomorfismi.
2. È stata definita l'**identità strutturale** di un sistema come **classe di isomorfismo globale**, indipendente da:
 - stati,
 - parametri,
 - esiti empirici.

3. È stata costruita una **firma strutturale canonica** $\text{Sig}(\Sigma)$ che rappresenta in modo:

- completo,
- normalizzato,
- verificabile,

l'identità strutturale del sistema.

4. È stato definito l'operatore **OSI** : $S \rightarrow \Omega$ come funzione:

- costruttiva,
- computabile,
- non circolare,

necessaria e sufficiente per la verifica dell'identità strutturale.

5. È stata introdotta una nozione di:

- equivalenza strutturale,
- distanza strutturale discreta,

coerente con la natura qualitativa dell'identità.

6. È stata formalizzata la classe delle **trasformazioni ammissibili**, chiusa sotto composizione, e caratterizzata in modo decidibile.

7. Sono stati dimostrati teoremi **non tautologici** riguardanti:

- preservazione delle proprietà strutturali dimostrabili,
- impossibilità del drift strutturale non osservabile,
- insufficienza dei criteri basati su esiti.

13.2 Stato di completezza del formalismo

Il formalismo presentato è **completo rispetto allo scopo dichiarato**.

In particolare:

- ogni nozione utilizzata è definita esplicitamente;
- ogni operatore è costruttivo;
- ogni equivalenza è verificabile;
- ogni teorema ha contenuto informativo non nullo.

Non rimangono:

- assunzioni implicite,
- dipendenze semantiche informali,
- richiami a concetti non formalizzati.

Dal punto di vista matematico, il nucleo PCIS è:

- **chiuso** (non richiede nuovi assiomi per il suo scopo),
- **consistente** rispetto alle definizioni adottate,
- **formalizzabile integralmente** in un proof assistant (es. Lean).

13.3 Relazione con opere derivate e applicazioni

Il presente trattato costituisce **opera fondazionale chiusa**.

Ogni opera futura:

- applicativa,
- ingegneristica,
- normativa,
- tecnologica,

potrà **derivare** dal PCIS, ma **non modificarlo**.

In particolare:

- non sono ammessi nuovi assiomi sostantivi all'interno del corpus PCIS;
- non sono ammesse estensioni che alterino il criterio di identità strutturale;
- ogni utilizzo applicativo deve rispettare i vincoli qui dimostrati.

13.4 Limiti dichiarati e non-pretese

È dichiarato esplicitamente che il PCIS:

- non fornisce criteri decisionali applicativi;
- non garantisce correttezza, sicurezza o optimalità;
- non predice comportamenti futuri;
- non sostituisce valutazioni empiriche, etiche o giuridiche.

Il suo scopo è **unico e preciso**:

fornire un criterio matematicamente rigoroso, necessario e sufficiente, per stabilire se l'identità strutturale di un sistema è stata conservata.

Ogni interpretazione oltre questo ambito è da considerarsi indebita.

14 . STRUCTURAL INVARIANCE MONITOR (SIM)

(Applicazione I — Prova atomica incontestabile)

14.1 Scopo dell'applicazione

Lo **Structural Invariance Monitor (SIM)** è un sistema operativo che realizza, in forma concreta, il criterio PCIS di conservazione dell'identità strutturale.

Il suo scopo è **rilevare in modo deterministico e non ambiguo** qualsiasi violazione dell'identità strutturale di un sistema, indipendentemente da:

- performance,
- esiti osservabili,
- parametri interni,
- stati temporanei.

SIM non valuta il sistema:

verifica se il sistema è ancora lo stesso.

14.2 Input formale

L'input di SIM è un sistema strutturale:

$$\Sigma = \langle F, G, H, V \rangle$$

espresso in una rappresentazione serializzata che preserva:

- insiemi finiti,
- relazioni,
- ordini,
- tipizzazione.

Forma minima accettata: **JSON strutturale canonico**.

Esempio (estratto concettuale, non numerico):

```
{  
  "F": { "roles": [...] },  
  "G": { "objectives": [...], "order": [...] },  
  "H": { "criteria": [...], "total_order": [...] },  
  "V": { "form": {...} }  
}
```

14.3 Operatore centrale

SIM implementa **direttamente** l'operatore:

$OSI : S \rightarrow \Omega$

Secondo la definizione formale:

$OSI(\Sigma) = \text{Sig}(\Sigma)$

dove $\text{Sig}(\Sigma)$ è la firma strutturale canonica.

Nessuna approssimazione è ammessa.

14.4 Architettura logica

SIM è composto da quattro moduli puramente funzionali:

1. Parser strutturale

- Converte input serializzato $\rightarrow$ strutture matematiche finite
- Verifica tipizzazione e completezza

2. Extractor

- Calcola:
 - σ_F
 - σ_G
 - σ_H
 - σ_V

3. Normalizer

- Canonicalizza le strutture a isomorfismo
- Elimina etichette irrilevanti

4. Comparator

- Confronta firme:
 - identiche $\rightarrow$ PASS
 - diverse $\rightarrow$ FAIL

Output binario.

14.5 Output

L'output di SIM è **discreto**:

IDENTITY_PRESERVED

oppure

IDENTITY_VIOLATED

Non esistono:

- gradi,
- percentuali,
- soglie.

Questo è imposto dal PCIS.

14.6 Pseudocodice minimale

```
function SIM_Check( $\Sigma$ _ref,  $\Sigma$ _cur):  
    sig_ref = OSI( $\Sigma$ _ref)  
    sig_cur = OSI( $\Sigma$ _cur)  
  
    if sig_ref == sig_cur:  
        return IDENTITY_PRESERVED  
    else:  
        return IDENTITY_VIOLATED
```

14.7 Implementazione Python minimale (replicabile)

```
def osi(system):  
    return canonical_signature(system)  
  
def sim_check(system_ref, system_cur):  
    return osi(system_ref) == osi(system_cur)
```

Nessuna dipendenza da ML, probabilità o metriche.

14.8 Vincoli hardware

SIM richiede esclusivamente:

- CPU standard
- memoria proporzionale alla dimensione finita delle strutture

- nessuna GPU
- nessuna accelerazione

È eseguibile:

- su laptop,
- server,
- embedded system.

14.9 Proprietà dimostrate (derivate da PCIS)

- Correttezza: segue dal Teorema 8.1
- Completezza: segue dal Teorema 8.2
- Immunità al drift non osservabile: Teorema 11.2

14.10 Stato dell'applicazione

SIM è:

- concettualmente completo,
- matematicamente determinato,
- implementabile riga per riga,
- replicabile da terzi **senza interpretazione**.

SIM **non è**:

- un sistema intelligente,
- un validatore di qualità,
- un controllore etico,
- un decisore.

È un **oracolo strutturale deterministico**.

15. STRUCTURAL REGENERATION FRAMEWORK (SRF)

(Applicazione II — Autorigenerazione isomorfa, non mutativa)

15.1 Scopo dell'applicazione

Lo **Structural Regeneration Framework (SRF)** realizza operativamente un concetto centrale del corpus PCIS:

un sistema può essere rigenerato solo se l'identità strutturale è preservata.

SRF non consente:

- auto-miglioramento,
- auto-ottimizzazione,
- auto-evoluzione,
- mutazione creativa.

SRF consente esclusivamente:

- ricostruzione,
- ripristino,
- duplicazione,
- ri-istanza,

a identità strutturale invariata.

15.2 Distinzione fondamentale: rigenerazione vs mutazione

Definizione 15.1 (Rigenerazione strutturale)

Una rigenerazione è una trasformazione:

$$T_{\text{reg}} : S \rightarrow S$$

tale che, per un sistema Σ :

$$\text{OSI}(T_{\text{reg}}(\Sigma)) = \text{OSI}(\Sigma)$$

La rigenerazione **non introduce** nuovi elementi strutturali.

Definizione 15.2 (Mutazione strutturale)

Una mutazione è una trasformazione:

$$T_{\text{mut}} : S \rightarrow S$$

tale che:

$$\text{OSI}(T_{\text{mut}}(\Sigma)) \neq \text{OSI}(\Sigma)$$

Ogni mutazione è **non ammissibile** nel framework PCIS.

15.3 Condizione necessaria e sufficiente di rigenerabilità

Teorema 15.1 (Condizione di rigenerazione isomorfa)

Un sistema Σ è rigenerabile **se e solo se** esiste un sistema Σ' tale che:

1. Σ' è costruibile da specifiche finite
2. $\text{OSI}(\Sigma') = \text{OSI}(\Sigma)$

Dimostrazione (schema rigoroso)

($\Rightarrow$)

Se Σ è rigenerabile, per definizione la rigenerazione è una trasformazione ammissibile.

Per Definizione 10.2, ogni trasformazione ammissibile conserva OSI.

($\Leftarrow$)

Se esiste Σ' con firma strutturale identica, allora per Teorema 8.2 vale $\Sigma' \cong \Sigma$.

La costruzione di Σ' è quindi una rigenerazione valida.

15.4 Architettura logica del framework SRF

SRF è composto da quattro livelli concettuali:

1. Structural Specification Layer

- definizione completa di F, G, H, V
- nessun parametro contingente

2. Canonical Signature Layer

- calcolo di $\text{Sig}(\Sigma)$

3. Regeneration Engine

- ricostruzione del sistema a partire dalla firma
- uso esclusivo di invarianti

4. Verification Gate

- controllo finale:
 $\text{OSI}(\Sigma_{\text{originale}}) == \text{OSI}(\Sigma_{\text{rigenerato}})$

Se il controllo fallisce, la rigenerazione è **rigettata**.

15.5 Algoritmo astratto di rigenerazione

Input: firma strutturale $\text{Sig}(\Sigma)$

Step 1: ricostruisci F' coerente con σ_F

Step 2: ricostruisci G' coerente con σ_G

Step 3: ricostruisci H' coerente con σ_H

Step 4: ricostruisci V' coerente con σ_V

$\Sigma' = \langle F', G', H', V' \rangle$

Se $\text{OSI}(\Sigma') == \text{Sig}(\Sigma)$:

ACCETTA

altrimenti:

RIGETTA

15.6 Implementazione Python minimale (concettuale)

```
def regenerate(signature):
```

```
    system_candidate = build_from_signature(signature)
```

```
    if osi(system_candidate) == signature:
```

```
        return system_candidate
```

```
    else:
```

```
        raise Exception("Structural violation")
```

Nota:

build_from_signature **non** è libero:

è una funzione deterministica vincolata dalla firma.

15.7 Proprietà garantite dal framework

Derivate **direttamente** dai teoremi PCIS:

- impossibilità di rigenerazione mutativa
- impossibilità di drift identitario
- replicabilità perfetta
- clonazione strutturale lecita
- ripristino dopo fault o crash

15.8 Cosa SRF NON consente (formalmente dimostrato)

SRF **non consente**:

- emergenza di nuove funzioni
- aumento di capacità cognitive
- auto-riscrittura strutturale
- “intelligenza crescente”
- AGI evolutiva
- perfezionamento indefinito

Tutto ciò richiederebbe:

$$\text{OSI}(\Sigma_{t+1}) \neq \text{OSI}(\Sigma_t)$$

che è vietato.

15.9 Relazione con sistemi biologici (nota interpretativa)

SRF **non simula** la biologia.

Formalizza invece un principio più forte:

la rigenerazione biologica reale è sempre isomorfa

la mutazione è sempre una rottura identitaria

Questa distinzione è **strutturale**, non semantica.

15.10 Stato dell'applicazione

SRF è:

- matematicamente dimostrato
- logicamente chiuso
- implementabile
- replicabile
- auditabile
- immune da interpretazioni “forti”

15.11 Risultato chiave

Autorigenerazione sì.

Auto-evoluzione no.

Questa è una **limitazione necessaria**, non una debolezza.

Transizione

Con **SIM (Sezione 14)** e **SRF (Sezione 15)** abbiamo dimostrato:

- identità verificabile

- rigenerazione controllata

Ora possiamo **scalare**.

16. PCIS-CONFORM MULTI-AGENT ARCHITECTURE (PMAA)

(Applicazione III — Scalabilità controllata senza perdita di identità)

16.1 Scopo dell'applicazione

La **PCIS-Conform Multi-Agent Architecture (PMAA)** risolve un problema strutturale noto e irrisolto:

come scalare sistemi cognitivi in più agenti senza perdere controllo identitario, senza emergenza non verificabile e senza mutazione strutturale.

PMAA non mira a:

- intelligenza collettiva emergente,
- super-intelligenza distribuita,
- auto-organizzazione non vincolata.

PMAA mira esclusivamente a:

- **composizione strutturalmente sicura,**
- **cooperazione verificabile,**
- **scalabilità senza drift.**

16.2 Definizione di agente PCIS-conforme

Definizione 16.1 (Agente strutturale)

Un agente PCIS-conforme è un sistema strutturale:

$$\Sigma_i = \langle F_i, G_i, H_i, V_i \rangle$$

che soddisfa integralmente le definizioni delle Sezioni 1–15.

Ogni agente è:

- strutturalmente chiuso,
- verificabile tramite OSI,
- non auto-modificante.

16.3 Spazio multi-agente

Definizione 16.2 (Sistema multi-agente PCIS)

Un sistema multi-agente è una collezione finita:

$$M = \{ \Sigma_1, \Sigma_2, \dots, \Sigma_n \}$$

con $n \geq 2$, tale che:

1. ogni Σ_i è PCIS-conforme;
2. ogni Σ_i ha firma strutturale definita;
3. le interazioni sono vincolate strutturalmente.

16.4 Vincolo fondamentale: nessuna identità collettiva

Assioma 16.1 (Assenza di identità emergente)

Non esiste alcun sistema strutturale globale Σ^M tale che:

$$\Sigma^M \neq \Sigma_i$$

e

OSI(Σ^M) sia diverso da ogni OSI(Σ_i)

In altre parole:

il sistema multi-agente non ha identità propria.

L'identità resta **atomica**, non collettiva.

16.5 Interazioni consentite tra agenti

Le interazioni sono modellate come **scambi strutturalmente neutri**.

Definizione 16.3 (Interazione ammessa)

Un'interazione tra due agenti Σ_i e Σ_j è una funzione:

$$I_{ij} : \text{Stato}_i \times \text{Stato}_j \rightarrow \text{Stato}_i \times \text{Stato}_j$$

tale che:

- non modifica F, G, H, V di nessun agente;
- agisce solo su stati contingenti.

Formalmente:

$$\text{OSI}(\Sigma_{i_post}) = \text{OSI}(\Sigma_{i_pre})$$

$$\text{OSI}(\Sigma_{j_post}) = \text{OSI}(\Sigma_{j_pre})$$

16.6 Comunicazione strutturalmente sicura

Definizione 16.4 (Canale PCIS-safe)

Un canale di comunicazione è PCIS-safe se:

1. trasmette solo dati tipizzati;
2. non introduce nuovi ruoli;
3. non altera obiettivi;
4. non modifica gerarchie decisionali;
5. non influisce sulla forma funzionale di valore.

16.7 Architettura logica PMAA

PMAA è composta da quattro livelli:

1. **Agent Layer**
agenti autonomi Σ_i
2. **Interaction Layer**
canali PCIS-safe
3. **Coordination Layer**
protocolli statici, non adattivi
4. **Audit Layer**
verifica OSI periodica o continua

Non esiste:

- livello meta-cognitivo,
- livello emergente,
- supervisore creativo.

16.8 Teorema centrale di scalabilità

Teorema 16.1 (Scalabilità senza drift)

Sia $M = \{\Sigma_1, \dots, \Sigma_n\}$ un sistema multi-agente PCIS-conforme.

Sia M' ottenuto aggiungendo un agente Σ_{n+1} tale che:

$OSI(\Sigma_{n+1}) = OSI(\Sigma_k)$ per qualche k ,
oppure Σ_{n+1} è un nuovo agente con firma valida.

Allora:

- nessun Σ_i cambia identità strutturale;
- M' resta PCIS-conforme.

Dimostrazione (schema)

Ogni agente è indipendente.

Le interazioni non modificano strutture.

OSI è invariante per ogni Σ_i .

Quindi nessun drift è possibile.

16.9 Impossibilità dell'intelligenza collettiva emergente

Teorema 16.2 (No emergent intelligence)

Non esiste alcuna proprietà strutturale P tale che:

$P(M)$ non sia riducibile a proprietà dei singoli Σ_i .

Dimostrazione:

- M non ha OSI;
- nessuna struttura globale è definita;
- quindi nessuna identità collettiva esiste.

16.10 Implementazione concettuale minimale

agents = [Σ_1 , Σ_2 , ..., Σ_N]

for agent in agents:

 assert osi(agent) == expected_signature(agent)

while system_running:

 exchange_state_data(agents)

 for agent in agents:

 assert osi(agent) == initial_signature(agent)

16.11 Proprietà garantite

PMAA garantisce:

- cooperazione controllata
- scalabilità orizzontale
- assenza di deriva
- auditabilità continua
- replicabilità totale
- fallimento localizzato (no contagio)

16.12 Cosa PMAA NON consente

Formalmente escluso:

- super-agente emergente
- coscienza collettiva
- auto-governo del sistema
- auto-evoluzione distribuita
- “mente globale”

Queste richiederebbero:

OSI(M) definibile

che è impossibile per costruzione.

16.13 Risultato

Si può scalare senza perdere controllo.

Ma non si può ottenere più identità dalla somma.

Questo chiude definitivamente il problema della **scalabilità sicura**.

Transizione

Ora il corpus ha dimostrato:

- identità verificabile (OSI)
- rigenerazione isomorfa (SRF)
- scalabilità controllata (PMAA)

SEZIONE 17 — AUDIT PACK & INSTITUTIONAL ADOPTION

(Applicazione IV — Verificabilità, governance e adozione formale)

17.1 Scopo della sezione

Questa sezione completa il corpus PCIS rendendolo:

- **adottabile** da istituzioni pubbliche e private,
- **verificabile** da terze parti indipendenti,
- **certificabile** senza fiducia soggettiva,
- **depositabile** come opera chiusa e opponibile.

Il risultato è un **Audit Pack strutturale completo**, non una proposta teorica.

17.2 Principio di auditabilità strutturale

Assioma 17.1 (Auditabilità totale)

Ogni affermazione rilevante nel framework PCIS deve essere:

- verificabile tramite OSI,
- indipendente da interpretazione semantica,
- riproducibile da un soggetto terzo.

Se una proprietà non è auditabile, **non è parte del sistema**.

17.3 Cosa si intende per Audit Pack

Definizione 17.1 (Audit Pack PCIS)

Un Audit Pack è un insieme finito di artefatti che consente a un revisore esterno di verificare:

1. l'identità strutturale di un sistema,
2. l'assenza di mutazioni non ammissibili,
3. la correttezza delle rigenerazioni,
4. la conformità multi-agente,
5. la stabilità nel tempo.

17.4 Contenuto minimo dell'Audit Pack

L'Audit Pack PCIS include **obbligatoriamente**:

1. **Specifica strutturale completa**
 - definizione esplicita di F, G, H, V
 - senza parametri nascosti
2. **Firma strutturale canonica**
 - output OSI
 - formato testuale deterministico
3. **Regole di ammissibilità**
 - definizione formale delle trasformazioni lecite
4. **Procedure di verifica**
 - istruzioni passo-passo
 - ripetibili da terzi

5. Log di controllo

- evidenza che OSI resta invariato

17.5 Audit come procedura deterministica

Definizione 17.2 (Procedura di audit)

Una procedura di audit è una funzione:

$\text{Audit} : \text{Artefatti} \rightarrow \{\text{PASS}, \text{FAIL}\}$

tale che:

- PASS se e solo se tutte le verifiche OSI coincidono,
- FAIL in presenza di una sola violazione strutturale.

Non esistono esiti intermedi.

17.6 Algoritmo di audit (astratto)

Input: sistema Σ , firma di riferimento Sig_ref

Step 1: calcola $\text{Sig_current} = \text{OSI}(\Sigma)$

Step 2: confronta Sig_current con Sig_ref

Step 3:

se uguali $\rightarrow$ PASS

altrimenti $\rightarrow$ FAIL

Questo algoritmo è:

- sufficiente,
- necessario,
- completo.

17.7 Governance senza discrezionalità

Teorema 17.1 (Governance strutturale)

Ogni sistema PCIS-conforme può essere governato **senza decisioni discrezionali**, usando esclusivamente:

- firme strutturali,
- regole di ammissibilità,
- audit periodico.

Dimostrazione.

Ogni violazione è rilevabile come differenza di OSI.
Nessuna valutazione soggettiva è necessaria.

17.8 Applicabilità istituzionale

Il framework è compatibile con:

- enti regolatori,
- sistemi di certificazione,
- ambienti industriali,
- infrastrutture critiche,
- contesti legali e normativi.

Perché:

- non richiede fiducia,
- non richiede segreti,
- non richiede interpretazione.

17.9 Deposito, priorità e opponibilità

Il corpus PCIS è progettato per:

- deposito in archivi pubblici,
- marcatura temporale opponibile,
- verifica post-hoc invariabile,
- protezione della priorità scientifica.

L'Audit Pack rende il contenuto **auto-verificante anche dopo anni**.

17.10 Limiti dichiarati dell'adozione

Il framework **non**:

- certifica qualità o performance,
- garantisce sicurezza operativa,
- sostituisce test empirici,
- assegna responsabilità legale automatica.

Certifica **solo e precisamente**:

la conservazione (o violazione) dell'identità strutturale.

17.11 Ruolo dell'IA in questo contesto

L'IA è utilizzabile esclusivamente come:

- strumento di implementazione,
- esecutore di procedure,
- supporto redazionale o tecnico.

L'IA **non è**:

- soggetto epistemico,

- titolare dell'opera,
- fonte di verità strutturale.

17.12 Chiusura applicativa

Con le Sezioni 14–17 il framework PCIS dispone ora di:

- prova atomica (Monitor),
- rigenerazione isomorfa (SRF),
- scalabilità controllata (PMAA),
- adozione istituzionale (Audit Pack).

Non manca **nulla** per l'adozione reale.

18 . CHIUSURA DEL CORPUS PCIS

18.1 Stato finale dell'opera

Il corpus PCIS è ora:

- formalmente chiuso,
- matematicamente rigoroso,
- epistemologicamente delimitato,
- tecnicamente implementabile,
- istituzionalmente adottabile.

18.2 Dichiarazione di chiusura

Non sono richieste:

- nuove sezioni,
- nuovi assiomi,
- nuove estensioni teoriche.

Ogni ulteriore lavoro è **derivato**, non fondazionale.

18.3 Dichiarazione di proprietà e diritti

L'intero corpus PCIS, incluse tutte le sezioni, applicazioni, formalizzazioni e testi:

© Pierernesto Di Paolo

Tutti i diritti riservati.

Uso, riproduzione, modifica o derivazione **vietati senza autorizzazione scritta**.

Contatto per autorizzazioni:

pierernesto.di.paolo.69@proton.me

18.4 Nota finale

Questo lavoro non promette “più intelligenza”.
Garantisce qualcosa di più raro:

controllo, verificabilità e limite dimostrato.